

UOS Pascal Interface to System Services

Table of contents

Title Page	3
Introduction	3
PasStarlet	3
ASCTIM	3
BINTIM	4
CRELNT	4
CVTIME	5
DAY_OF_WEEK	6
DELLNM	6
DEVICE_SCAN	7
FAO and FAOL	7
FILESCAN	8
GET_COMMAND	9
Get_Default_File_Protection	9
Get_Input	10
Get_Recall	10
Get_Symbol	11
Get_Wildcard_Symbol	11
GETDIR	12
GETDVIW	12
GETJPIW	13
GETMSG	14
GETSYIW	14
GETUAI	15
Put_Formatted_Output	16
Put_Output	16
RUN	17
Set_Symbol	18
SPAWN	18
Substitute_Wildcards	19
TRNLNM	20
LBRPas	21
Close	21
Close_Control	21
Delete	22
Get_Module	22
Ini_Control	23
Last_Error	24
Module_Exists	24
Open	25
Output_Help	25
Put_Module	26

Title Page

UOS

Pascal Interface to System Services

October 2023

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

Introduction

Introduction

This document is intended for programmers using Pascal to write programs to run on UOS.

PasStarlet.pas and LBRPas are Pascal source files that contain functions that simplify the use of UOS system services. There are three means used to provide this simplification:

- Allow passing and return of native Pascal strings rather than having to create TSRB structures.
- Allows passing values rather than pointer to values.
- Returns result values rather than condition codes. The consequence of this is that the caller must specifically check for exceptions after the call. Note: Some interfaces that return no other data will return a success code.

Not all system services are suitable for the use of the Pascal interfaces, such as those requiring no parameters and returning no strings.

These interfaces are provided in source form and use standard Pascal so that they will work with any Pascal compiler.

Each interface function is described briefly, with a reference to the system service and a function prototype. The documentation for the corresponding system service should be consulted for a detailed description of the meaning of parameters.

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

PasStarlet

PasStarlet.pas contains interfaces for the LIB\$ and SYS\$ services.

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

ASCTIM

ASCTIM

Interface for: SYS\$ASCTIM

Definition:

```
function ASCTIM( Timestamp : int64 ; Flags : integer = 0 ) : string ;
```

Parameters:

Timestamp
Timestamp to convert to text.

Flags
Options for processing.

Description:

Returns a human-readable representation of the passed timestamp.

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

BINTIM

BINTIM

Interface for: SYS\$BINTIM

Definition:

```
function BINTIM( const time : string ) : int64 ;
```

Parameters:

time
Textual time to convert to a timestamp.

Description:

Returns a timestamp representing the passed date/time string.

Created with the Personal Edition of HelpNDoc: [Free Web Help generator](#)

CRELNT

CRELNT

Interface for: SYS\$ASCTIM

Definition:

```
function CRELNT( Attr : int64 ; var Resnam : string ; Quota : int64 ; Promsk : int64 ; Tabnam,  
Partab : string ; AcMode : int64 ) : int64 ;
```

Parameters:

Attr
Attributes for the table.

Resnam
Name of table on return.

Quota
Table quota, in bytes.

Promsk
Protection mask for table.

Tabnam
Name for table.

Partab
Name of parent table.

AcMode
Access Mode for table.

Description:

Creates a symbol table. Returns a condition code.

Created with the Personal Edition of HelpNDoc: [Easily create Web Help sites](#)

CVTIME

CVTIME

Interface for: LIB\$CVTIME

Definition:

```
function CVTime( var Time : string ; Format, Output : word ) : cardinal ;
```

Parameters:

Time
Textual time to convert on entry, the converted time upon return.

Format

How to format the result

Output

Which items to return.

Description:

Returns a time representing the passed date/time string with the appropriate format and output options. The time is returned in the first parameter. The function returns a condition code.

Created with the Personal Edition of HelpNDoc: [Easy Qt Help documentation editor](#)

DAY_OF_WEEK

DAY_OF_WEEK

Interface for: LIB\$DAY_OF_WEEK

Definition:

```
function DAY_OF_WEEK( Time : int64 ) : integer ;
```

Parameters:

Time

Timestamp for which to return day of week.

Description:

Returns a value indicating the day of the week for the passed timestamp.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

DELLNM

DELLNM

Interface for: SYS\$DELLNM

Definition:

```
function DELLNM( Table, Symbol : string ; ACMMode : integer = 4 ) : int64 ;
```

Parameters:

Table

Timestamp for which to return day of week.

Symbol

Name of symbol to delete.

ACMode

Access mode for the symbol.

Description:

Deletes a symbol from a table. Returns a condition code.

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

DEVICE_SCAN

DEVICE_SCAN

Interface for: SYS\$DEVICE_SCAN

Definition:

```
function DEVICE_SCAN( const Devnam : string ; Itemlist, Context : int64 ) : string ;
```

Parameters:

Devnam

Name of device.

Itemlist

Pointer to a list of items.

Context

Context from previous and for subsequent calls.

Description:

Iterates through devices, returning data.

Created with the Personal Edition of HelpNDoc: [Free Kindle producer](#)

FAO and FAOL

FAO and FAOL

Interface for: SYS\$FAO and SYS\$FAOL

Definition:

```
function FAO( Control : string ; P1 : int64 = 0 ;
  P2 : int64 = 0 ; P3 : int64 = 0 ; P4 : int64 = 0 ; P5 : int64 = 0 ;
  P6 : int64 = 0 ; P7 : int64 = 0 ; P8 : int64 = 0 ; P9 : int64 = 0 ;
  P10 : int64 = 0 ; P11 : int64 = 0 ; P12 : int64 = 0 ; P13 : int64 = 0 ;
  P14 : int64 = 0 ; P15 : int64 = 0 ; P16 : int64 = 0 ; P17 : int64 = 0 ) : string ;

function FAOL( Control : string ; Parameters : int64 = 0 ) : string ;
```

Parameters:

Control
Text to format.

Parameters
Pointer to a list of parameters

P1..P17
Parameters

Description:

Formats text.

Created with the Personal Edition of HelpNDoc: [Free Web Help generator](#)

FILESCAN

FILESCAN

Interface for: SYS\$FILESCAN

Definition:

```
function FILESCAN( const Name : string ) : TStringList ;
```

Parameters:

Name
File name to scan.

Description:

Returns a list of fields from the passed filename. The returned string list contains the following items:

Offset	Description
--------	-------------

0	Node name
1	Node access control string
2	Primary node name
3	Secondary node information
4	Device name
5	Root directory
6	Directory name
7	File name
8	File type
9	Version

Any fields missing from the specification will be null in the list.

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

GET_COMMAND

GET_COMMAND

Interface for: SYS\$GET_COMMAND

Definition:

```
function Get_Command( const Prompt : string ) : string ;
```

Parameters:

Prompt
Text to prompt user with.

Description:

Returns input from sys\$command.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

Get_Default_File_Protection

Get_Default_File_Protection

Interface for: LIB\$Get_Default_File_Protection

Definition:

```
function Get_Default_File_Protection( PID : TPID ) : int64 ;
```

Parameters:

PID
Process ID whose file protection is to be returned. 0 for current process.

Description:

Returns the default file protection flag for the specified process.

Created with the Personal Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

Get_Input

Get_Input

Interface for: LIB\$Get_Input

Definition:

```
function Get_Input( const Prompt : string ) : string ;
```

Parameters:

Prompt
Text to prompt user with.

Description:

Returns input from user.

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

Get_Recall

Get_Recall

Interface for: LIB\$Get_Recall

Definition:

```
function Get_Recall( Index : integer ) : string ;
```

Parameters:

Index
Index of item in recall buffer.

Description:

Returns the specified line from the recall buffer.

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

Get_Symbol

Get_Symbol

Interface for: LIB\$Get_Symbol

Definition:

```
function Get_Symbol( const Name : string ; Table : longint = 0 ) : string ;
```

Parameters:

Name

Name of the symbol.

Table

Table to search for symbol.

Description:

Return a symbol value with no translation. Use TRNLNM to translate symbols.

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

Get_Wildcard_Symbol

Get_Wildcard_Symbol

Interface for: LIB\$Get_Wildcard_Symbol

Definition:

```
function Get_Wildcard_Symbol( const Name : string ; Table : longint = 0 ) : string ;
```

Parameters:

Name

Name of the symbol.

Table

Table to search for symbol.

Description:

Return a symbol value with no translation. Use TRNLNM to translate symbols. The symbol name can include wildcards. The first matching symbol is returned.

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

GETDIR

GETDIR

Interface for: SYS\$GETDIR

Definition:

```
function GETDIR : string ;
```

Description:

Returns the current default directory.

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

GETDVIW

GETDVIW

Interface for: SYS\$GETDVIW

Definition:

```
procedure GETDVIW( efn, Handle : int64 ; const Device : string ; iitmlist, IOSB, ASTAdr, ASTPrm : int64 ; const Pathname : string ; Siz : integer ) ;
```

Parameters:

efn
Event flags

Handle
Handle to open device. If 0, the device name is used.

Device
Name of device for which to return information. Ignored if handle passed.

itmlist
Pointer to list of descriptors indicating what to return.

IOSB

IO Status block

ASTAdr

Address for return of information.

ASTPrm

Parameter to return for ASTAdr routine.

Pathname

Pathname for multipath devices.

Siz

Number of descriptors in array.

Description:

Returns information on the specified device.

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

GETJPIW

GETJPIW

Interface for: SYS\$GETJPIW

Definition:

```
procedure GETJPIW( efn, pidadr : int64 ; const prcnam : string ; itm1st, iosb, astadr, astprm : int64 ) ;
```

Parameters:

efn

Event flags

pidadr

Address of PID

prcnam

Name of device for which to return information. Ignored if handle passed.

itm1st

Pointer to list of descriptors indicating what to return.

IOSB

IO Status block

ASTAdr
Address for return of information.

ASTPrm
Parameter to return for ASTAdr routine.

Description:

Returns information on the specified process.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

GETMSG

GETMSG

Interface for: LIB\$SYS_GETMSG

Definition:

```
function GETMSG( Msg, Flags : int64 ) : string ;
```

Parameters:

Msg
Pointer to message number to return.

Flags
Pointer to flags indicating which items to return.

Description:

Returns the text of the specified message number.

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

GETSYIW

GETSYIW

Interface for: SYS\$GETSYIW

Definition:

```
procedure GETSYIW( efn, CSIAdr : int64 ; Name : string ; ltm1st, IOSB, astadr, astprm : int64 ) ;
```

Parameters:

efn

Event flags

csiadr

Address of cluster ID

Name

Name of node for which to return information. Ignored if handle passed.

itmlist

Pointer to list of descriptors indicating what to return.

IOSB

IO Status block

ASTAdr

Address for return of information.

ASTPrm

Parameter to return for ASTAdr routine.

Description:

Returns information on the specified (or current) system.

Created with the Personal Edition of HelpNDoc: [Free Kindle producer](#)

GETUAI

GETUAI

Interface for: SYS\$GETUAI

Definition:

```
function GETUAI( context : int64 ; const username : string ; itmelist : int64 ) : int64 ;
```

Parameters:

context

Context for lookups.

username

Name of user for whom to return information.

itmlist

Pointer to list of descriptors indicating what to return.

Description:

Returns information on the specified user.

Created with the Personal Edition of HelpNDoc: [Free Web Help generator](#)

Put_Formatted_Output

Put_Formatted_Output

Interface for: LIB\$Put_Formatted_Output

Definition:

```
function Put_Formatted_Output( F : TCOM_UOS_File ; const Text : string ) : int64 ;
```

Parameters:

F

File to write to.

Text

Text to output to file.

Description:

Writes text to the passed file.

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

Put_Output

Put_Output

Interface for: LIB\$Put_Output

Definition:

```
function Put_Output( Text : string ) : int64 ;
```


Parameters:

Text
Text to output to sys\$output.

Description:

Writes text to sys\$output.

Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

RUN

RUN

Interface for: LIB\$RUN

Definition:

```
function RUN( _Program, Command_Line : string ; Accounting : boolean = True ; Authorize : boolean = True ; Debug : boolean = False ; Dump : boolean = False ) : int64 ;
```

Parameters:

_Program
File specification of program to run.

Command_Line
Command line to pass to program.

Accounting
True to run with accounting.

Authorize
True to use normal authorization.

Debug
True to run in debug mode.

Dump
True to do a memory dump when image ends.

Description:

Runs a program. Returns a condition code.

Set_Symbol

Set_Symbol

Interface for: LIB\$Set_Symbol

Definition:

```
function Set_Symbol( const Name, Value : string ; Table : longint = 0 ) : longint ;
```

Parameters:

Name

Name of the symbol.

Value

Value to assign to symbol.

Table

Table to search for symbol.

Description:

Set the value of a symbol. Returns the success code.

SPAWN

SPAWN

Interface for: LIB\$SPAWN

Definition:

```
function SPAWN( Command : string ; _Input : string ; _Output : string ; Flags : integer ; Name : string ;  
PID : integer; var Status : int64 ; Event_Flags : int64 = 0 ; AST : int64 = 0 ; ASTPm : int64 = 0 ;  
Prompt : string = " ; CLI : string = " ; Table : string = " ) : int64 ;
```

Parameters:

Command

File to execute in subprocess

_Input

Input device name

_Output
Output device name

Flags
Process creation flags.

Name
Name to assign to subprocess.

PID
Address to receive PID of subprocess.

Status
Contains status on return.

Event_Flags
Event flags.

AST
Address of AST.

ASTPrm
Parameter to pass to AST.

Prompt
Prompt to use.

CLI
Shell to use.

Table
Symbol table to use

Description:

Creates a subprocess.

Substitute_Wildcards

Substitute_Wildcards

Interface for: LIB\$Substitute_Wildcards

Definition:

```
function Substitute_Wildcards( Source, Target, Defaults : string ) : string ;
```

Parameters:

Source

Source filename.

Target

Destination filename, which may contain wildcards that are replaced from Source.

Defaults

File specification containing fields that will be substituted for missing fields in Source.

Description:

Return a file specification, substituting wildcards with defaults and values from a non-wildcard specification.

Created with the Personal Edition of HelpNDoc: [Free help authoring tool](#)

TRNLNM

TRNLNM

Interface for: SYS\$TRNLNM

Definition:

```
function TRNLNM( attr : int64 ; tabnam, lognam : string ; acmode, itmlst : int64 ) : int64 ;
```

Parameters:

attr

Context for lookups.

tabnam

Table name.

lognam

Symbol name.

acmode

Access mode.

itmlst

Address of descriptor list indicating what information to return.

Description:

Returns a symbol value and/or metadata.

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

LBRPas

LBRPas.pas contains Pascal interfaces for the LBR services.

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

Close

CLOSE

Interface for: LBR\$CLOSE

Definition:

```
function Close( context : int64 ) : int64 ;
```

Parameters:

context
Context from INI_Control

Description:

Closes a library.

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

Close_Control

CLOSE_CONTROL

Interface for: LBR\$CLOSE_CONTROL

Definition:

```
function Close_Control( Context : int64 ) : int64 ;
```

Parameters:

context
Context from INI_Control.

Description:

Closes a context.

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

Delete

DELETE

Interface for: LBR\$DELETE

Definition:

```
function Delete( context : int64 ; name : string ) : int64 ;
```

Parameters:

context
Context from INI_Control

name
Name of module to delete.

Description:

Deletes a module from a library.

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

Get_Module

GET_MODULE

Interface for: LBR\$GET_MODULE

Definition:

```
function Get_Module( context : int64 ; routine : int64 = 0 ; data : int64 = 0 ; key : string = " ) : int64 ;
```

Parameters:

context

Context from INI_Control

routine

Address of callback routine.

data

Value to pass to callback routine.

key

Name of the module to return.

Description:

Obtains a module from a library.

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

Ini_Control

LBR_INI_CONTROL

Interface for: LBR\$INI_CONTROL

Definition:

```
function Ini_Control( var c : int64 ; func, typ : int64 ; name, default : string ; var Res : string ) : int64 ;
```

Parameters:

c

Context

func

Access type

typ

Library type

name

Filename of the library

default

Default name of library, missing fields in name are filled from this.

Res

On return, contains the name of the library, as constructed from name and default.

Description:

Prepares a library file to be opened.

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

Last_Error

Last_Error

Definition:

```
function Last_Error : int64 ;
```

Parameters:

Description:

Returns the last exception that occurred when using an LBR service.

Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

Module_Exists

MODULE_EXISTS

Interface for: LBR\$MODULE_EXISTS

Definition:

```
function Module_Exists( Context : int64 ; Name : string ) : boolean ;
```

Parameters:

Context
Context from INI_Control

Name
Name of the module to check.

Description:

Returns true if the specified module exists in the library.

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

Open

OPEN

Interface for: LBR\$OPEN

Definition:

```
function Open( context : int64 ; Options : array of int64 ; name : string = " ) : int64 ;
```

Parameters:

context
Context from INI_Control

Options
Creation options

name
Name of the library to open. If null, the name used with INI_Control is used.

Description:

Opens a library for access.

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

Output_Help

OUTPUT_HELP

Interface for: LBR\$OUTPUT_HELP

Definition:

```
function Output_Help( routine : int64 ; width : int64 = 0 ; path : string = " ; lib : string = " ; flags : int64 = 0 ; input : int64 = 0 ) : int64 ;
```

Parameters:

routine
Callback for showing help text.

width
Width indicating position where to wrap text on each line. 0 indicates that it depends upon the output device.

path
Path of help module to output.

lib
File specification of help library

flags
Options.

input
Optional input callback.

Description:

Displays help text from a help library.

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

Put_Module

PUT_MODULE

Interface for: LBR\$PUT_MODULE

Definition:

```
function Put_Module( context : int64 ; name : string ; const module : string ) : int64 ;
```

Parameters:

context
Context from INI_Control

name
Name of the module to write to the library.

module
Contents of module.

Description:

Creates a module and writes data to it.

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)
